The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Ex parte DAVID ROBERT BALDWIN

Appeal No. 2005-1995
Application 09/133,741[1]

HEARD: December 14, 2005

Before KRASS, BARRETT, and DIXON, Administrative Patent Judges.

BARRETT, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-3, 7-18, 20-27, and 47-52.

We reverse.

[1] Application for patent filed August 13, 1998, entitled "Triangle Clipping for 3D Graphics," which is based on and claims priority under 35 U.S.C. § 119(e)(1) from U.S. Provisional Application 60/071,590, filed January 15, 1998.

BACKGROUND

The invention relates to method for clipping which uses a circular buffer instead of two separate buffers which are ping-ponged between (specification, pages 11-13).

Claim 1 is reproduced below.

> 1. A method for clipping graphics primitives for display, comprising the steps of:
>
> performing a clipping algorithm which uses only a single circular buffer to store input and output vertices of a primitive; and
>
> for each one of said vertices, indicating whether said one of said vertices is visible with respect to each plane of a view volume.

THE REFERENCES

The examiner relies on the following references:

Watkins et al. (Watkins)	5,361,386	November 1, 1994
Narayanaswami	5,613,052	March 18, 1997
Rossin et al. (Rossin)	5,877,773	March 2, 1999 (filed May 30, 1997)

Ivan E. Sutherland, Micropipelines, Communications of the ACM, Volume 32, Number 6, pp. 720-738, June 1989.

THE REJECTIONS

Claims 1, 2, 7-11, 14, 15, 48, 50, and 51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Rossin and Sutherland.

Claims 16, 17, 20-24, 26, 27, and 47 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Rossin and Sutherland, further in view of Watkins.

Claims 3, 12, 13, 49, and 52 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Rossin and Sutherland, further in view of Narayanaswami.

Claims 18 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Rossin and Sutherland, further in view of Watkins and Narayanaswami.

We refer to the final rejection (pages referred to as "FR__") and the examiner's answer (pages referred to as "EA__") for a statement of the examiner's rejection, and to the brief (pages referred to as "Br__") and reply brief (pages referred to as "RBr__") for a statement of appellant's arguments thereagainst.

OPINION

The examiner finds (EA3-4):

> Rossin et al teaches a method for clipping graphics primitives. The method comprises the steps of: using a clipping algorithm with only one buffer 304 (Fig. 3 and 5A) to store input and output polygons of the primitive (fig. 5A; col. 8, lines 66-67 and col. 11, lines 48-51; col. 12, lines 4-12; and col. 4, lines 17-32); and indicating whether each vertices [sic] is visible in each plane (col. 7, lines 58-65).

The examiner finds that Rossin does not disclose using a circular buffer, but finds that "Sutherland suggests using a circular buffer to store clip data (page 720, last two lines of the first column; page 732, second column, section 'Other Devices using the Same Protocol'; page 735, first column, first paragraph)" (EA4), and concludes that it would have been obvious to replace the one

buffer of Rossin with the circular buffer of Sutherland "to built [sic] a simple storage device capable or [sic] replacing oldest data with the newest data in clipping operation in which vector length are always changing" (EA4).

Appellant argues that none of the references alone or in combination teach or suggest the limitation of using "only a single circular buffer" to store input and output vertices of a primitive, as recited in independent claims 1, 16, and 48.

The examiner responds that Rossin's buffer 304 is the only memory that stores both input and output vertices and the other buffer 306 does not store both input and output vertices of a primitive (EA9). The examiner finds that since Rossin "can shift data from one memory location (the output list location) to another location (the input list location) ..., this clearly suggests a characteristic of a circular buffer" (EA11-12) which suggests substitution of the ring buffer in Sutherland (EA12).

Appellant replies that Rossin requires an additional memory device 306 in order to perform the clipping process and, therefore, does not meet the limitation of using "only a single circular buffer" to store input and output vertices of a primitive (RBr2). It is also argued that shifting vertices from the input list location to the output list location does not suggest a circular buffer (RBr2-3).

We agree with appellant on both arguments. The vertex look up table (VLUT) 304 in Figs. 3 and 5A stores input and output vertices of a primitive. However, the vertex RAM 306 also stores input and output vertices of the primitive (see col. 11, lines 59-62: "the first four memory locations of VRAM 306 store the input vertices of the input primitive, and are not overwritten during ... clipping," and col. 12, lines 8-10: "The clipper 308 ... writes the output vertices into the VRAM 306), although not in an ordered list as in VLUT 304. Therefore, the circuit does not use only a single buffer to store input and output vertices as claimed.

The operation of the VLUT 304 does not suggest use of a circular (ring) buffer. A circular buffer is a first-in, first-out (FIFO) buffer that it uses head and tail pointers to indicate write and read memory locations. The memory locations "wrap" around to create a circular list of addresses. Data is added to the location pointed to by the write pointer and is read from the location pointed to by the read pointer. The VLUT 304 does not operate in this manner. Referring to Fig. 5A, "[w]hen the clipper 308 is called, the vertices in the output list are shifted into the corresponding locations in the input list and the output list is cleared" (col. 12, lines 4-6). This does not act like a circular buffer. Rossin indicates that "[t]he VLUT 304 is preferably implemented in a double buffered RAM" (col. 11,

lines 46-47). Assume there are two buffers, Buffer 1 and Buffer 2. The output list of vertices from the Initial State in Buffer 1 is copied to the input list in (initially cleared) Buffer 2; Buffer 1 is cleared; the clipping routine generates vertices in the output list of Buffer 2; the output list from Buffer 2 is copied to the input list is Buffer 1; Buffer 2 is cleared; and the process repeats. It is possible to copy the output list to the input list by overwriting the input list, but this also does not act like a circular buffer.

Sutherland states (page 735, first col.):

> Perhaps the most important applications of micropipelines will involve operations in which the vector length changes. One such example is the clipping operation widely used in computer graphics. The clipping operation removes the parts of a set of objects that lie outside a reference window. Clipping may result in an increase or decrease in the number of objects in the set. Because whole objects may be removed, there may be less output than input, but because connected edges may also be broken into multiple pieces, there may also be more output than input. Such a clipping device with very simple interface characteristics can be built using the micropipeline framework.

Although appellant states that "[t]his passage certainly does suggest the use of circular buffers in connection with clipping operations, but does not suggest COMBINING two conventional circular buffers into one" (Br12), we find no mention of circular buffers here. However, Sutherland earlier stated that "one can build a ring-buffer FIFO whose interface characteristics are the same as those of the micropipelined FIFOs of Figure 15 or Figure 16" (page 732), and this suggests that micropipelines, linear or

circular, can be used for clipping. However, Sutherland (the same Sutherland as in the Sutherland-Hodgman clipping algorithm) contains no teaching of how clipping is to be done and does not disclose clipping using a circular buffer.

For the reasons stated above, we find no teaching or suggestion in either Rossin or Sutherland to use "only a single circular buffer" to store input and output vertices of a primitive. Since the limitation of "only a single circular buffer" is in all claims, and since the other references are not relied upon for this teaching, the rejections of claims 1-3, 7-18, 20-27, and 47-52 are reversed.

REVERSED

ERROL A. KRASS)
Administrative Patent Judge)
)
)
)
LEE E. BARRETT) BOARD OF PATENT
Administrative Patent Judge) APPEALS
) AND
) INTERFERENCES
)
)
)
JOSEPH L. DIXON)
Administrative Patent Judge)

Appeal No. 2005-1995
Application 09/133,741

GROOVER & HOLMES
BOX 802889
DALLAS, TX 75380-2889